

(12) UK Patent Application (19) GB (11) 2 317 302 (13) A

(43) Date of A Publication 18.03.1998

(21) Application No 9619043.4

(22) Date of Filing 12.09.1996

(71) Applicant(s)
Sharp Kabushiki Kaisha

(Incorporated in Japan)

22-22 Nagaike-cho, Abeno-ku, Osaka-shi, Osaka 545,
Japan

(72) Inventor(s)
Paul Joseph Kearney
Robert Maxwell Smith
Arvindra Singh Sehmi

(74) Agent and/or Address for Service
Marks & Clerk
Alpha Tower, Suffolk Street Queensway,
BIRMINGHAM, B1 1TT, United Kingdom

(51) INT CL⁶
G06F 17/60, H04Q 3/00

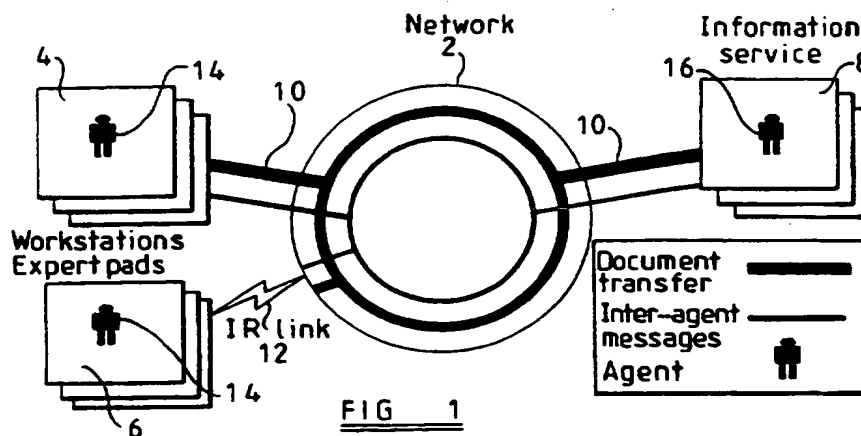
(52) UK CL (Edition P)
H4K KF42 KOD4
G4A AUDB

(56) Documents Cited
GB 2298299 A GB 2294788 A GB 2294132 A
WO 96/25012 A1 WO 96/23265 A1 WO 96/18963 A1
WO 96/05563 A1 WO 95/30317 A1 US 5305200 A
US 5202827 A

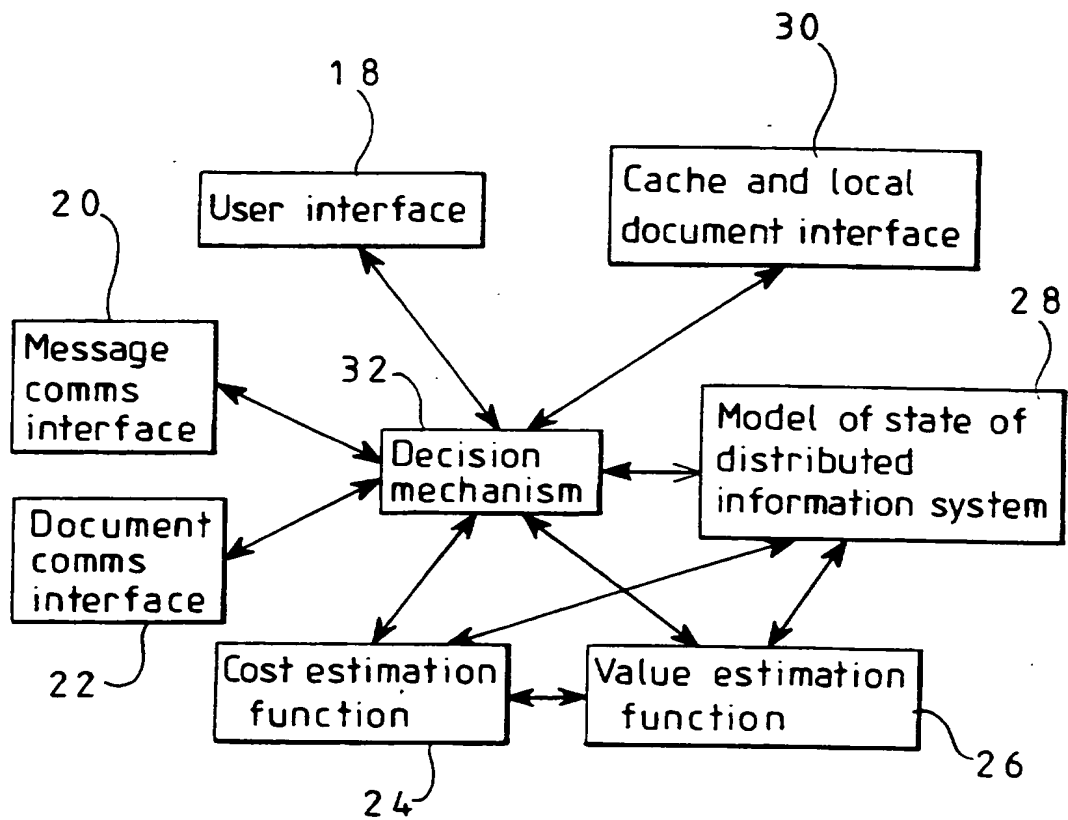
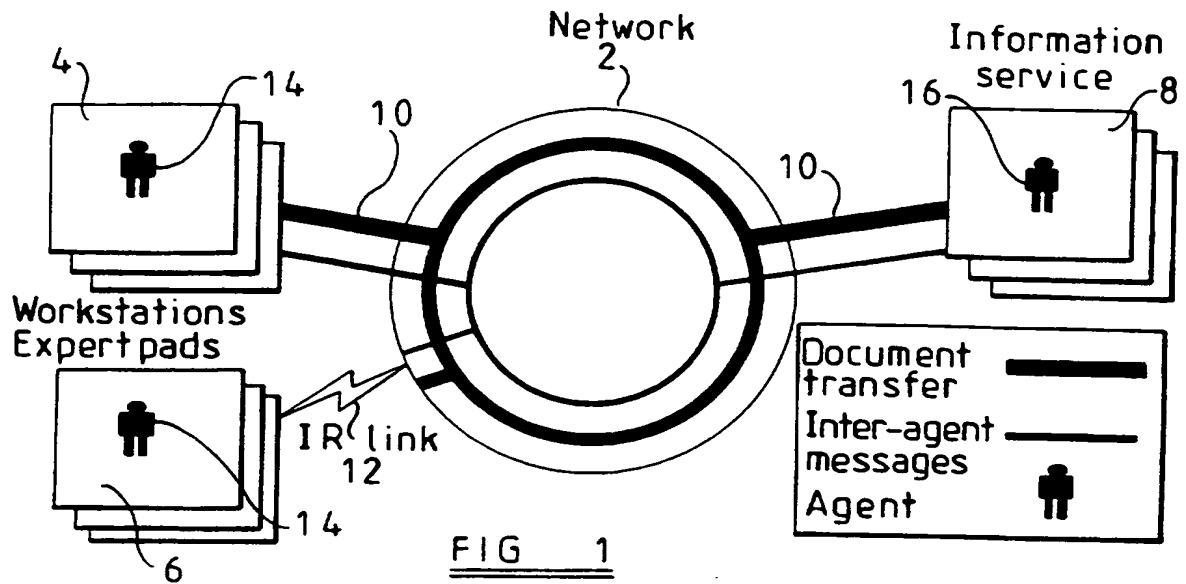
(58) Field of Search
UK CL (Edition O) G4A AUDB AUXF, H4K KF42 KOD4
KOT
INT CL⁶ G06F, H04Q
ONLINE: WPI

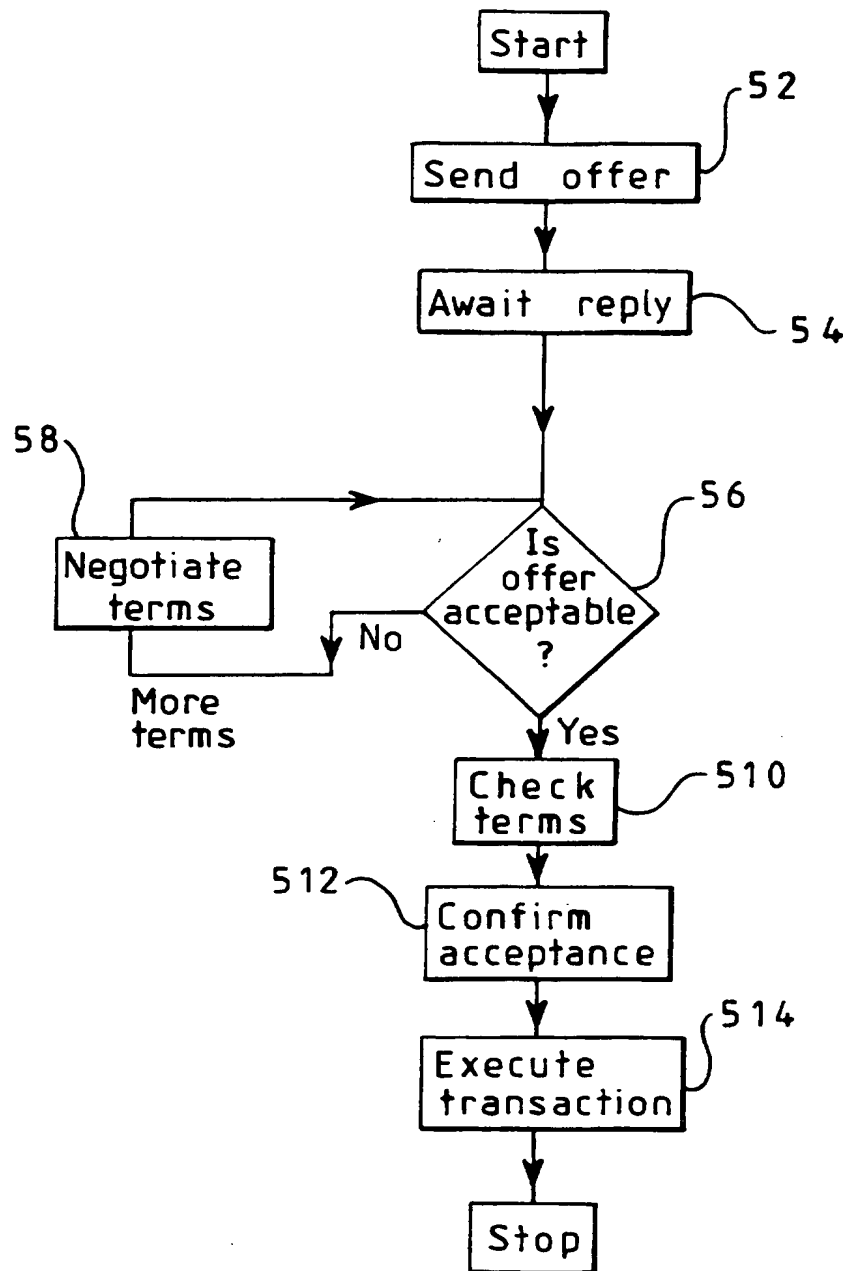
(54) A distributed information system

(57) A distributed information system comprises information containing sites (8) connected to a network (2). A network user software agent (4) has means (24, 26, 28) for determining the value of the system in response to a proposed modification of the distribution of the information in the system. For instance, such a modification may comprise moving the information from one site (8) to another, storing a copy in a cache (30) or deleting a copy from the cache (30). The user (4) institutes the proposed modification if the value exceeds a threshold.



1 / 13



FIG 3

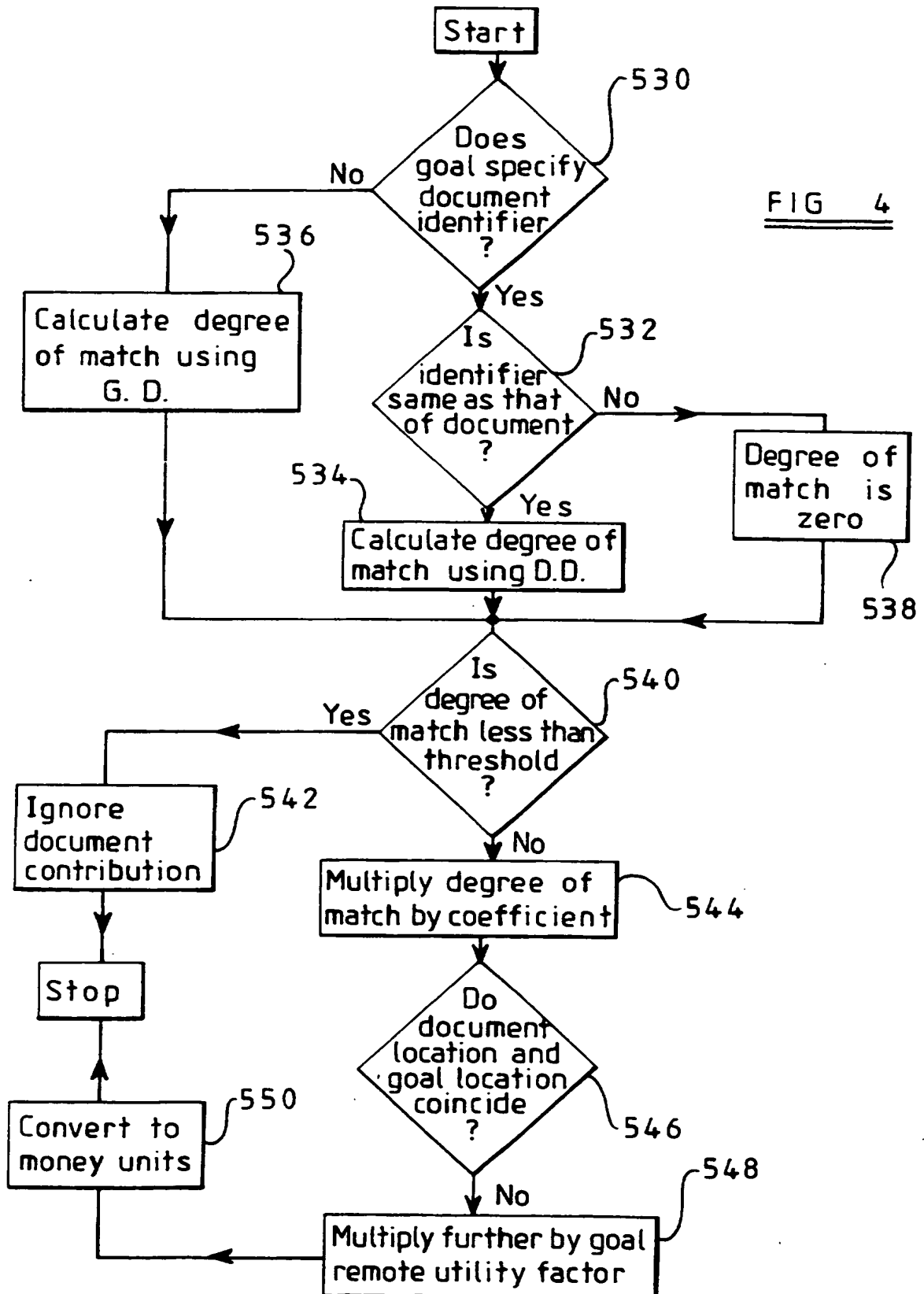


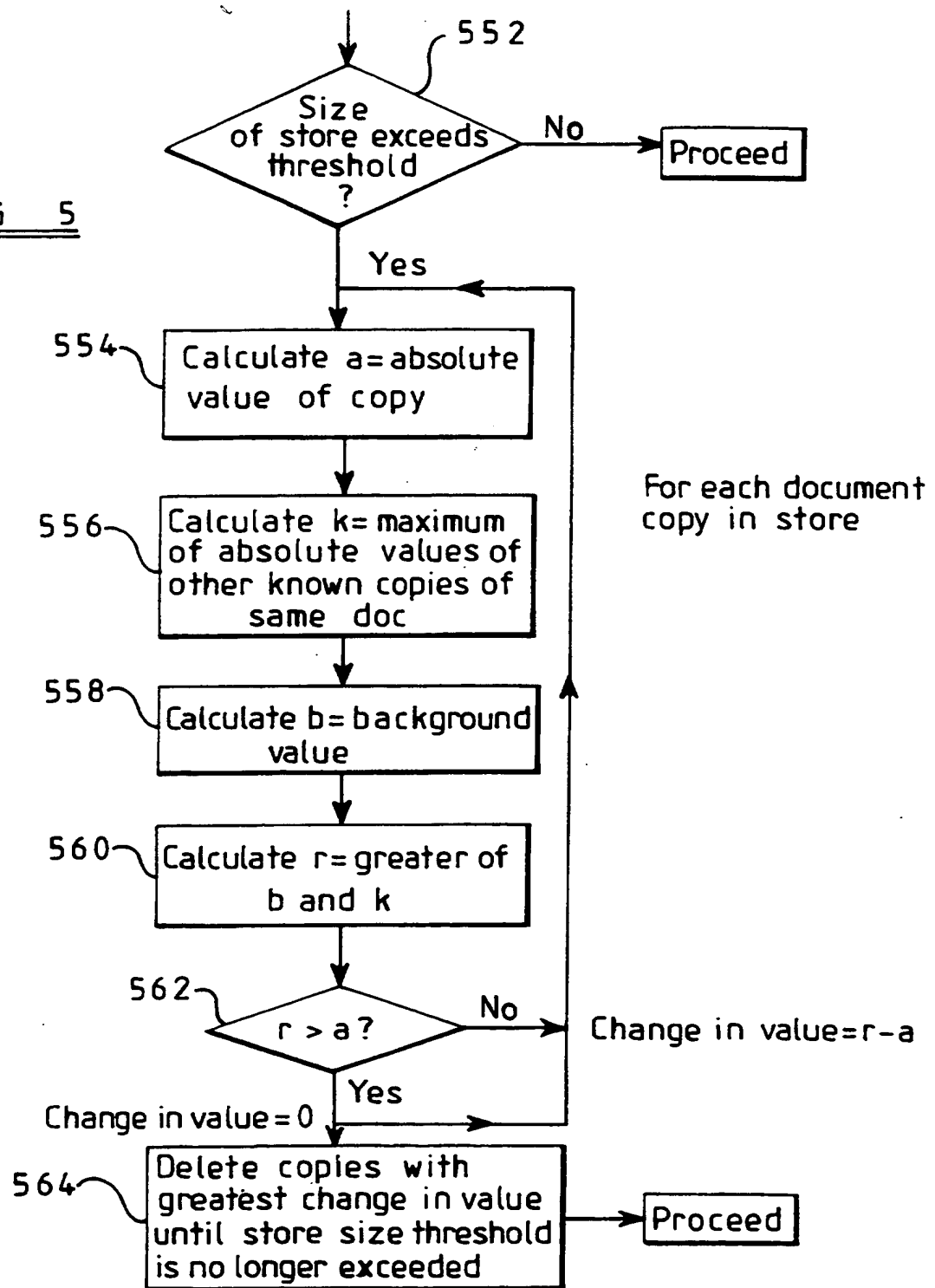
FIG 5

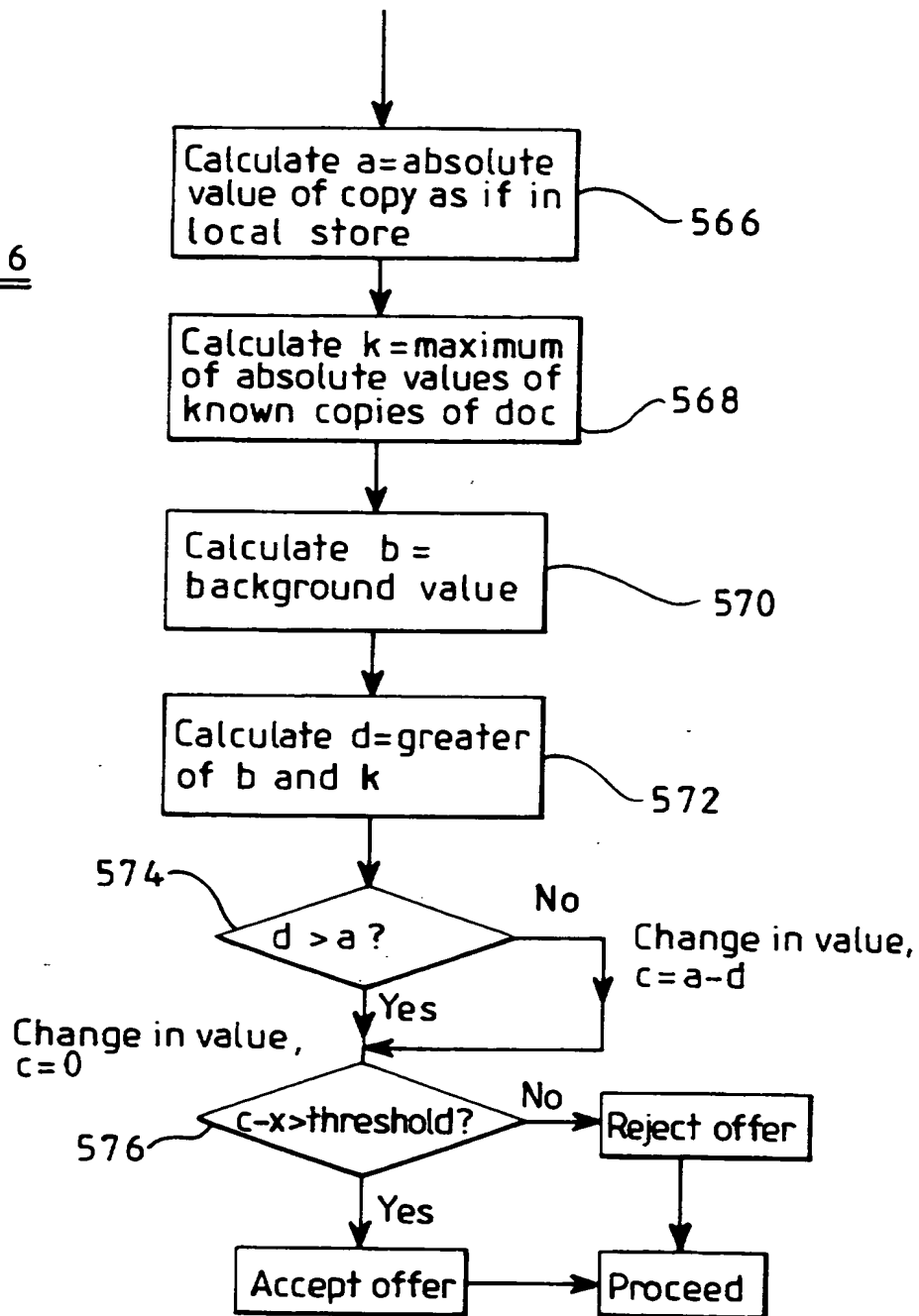
FIG 6

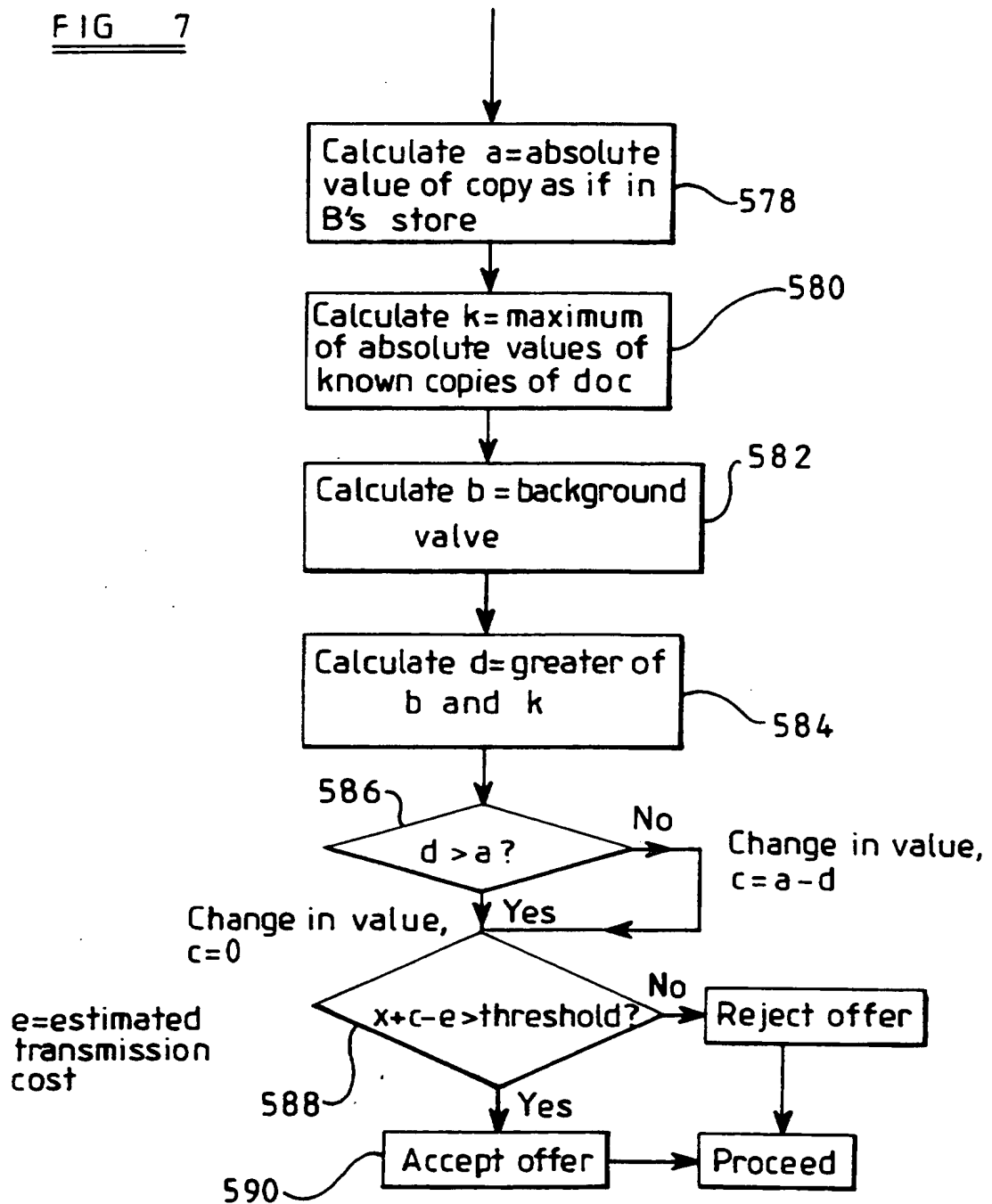
FIG 7

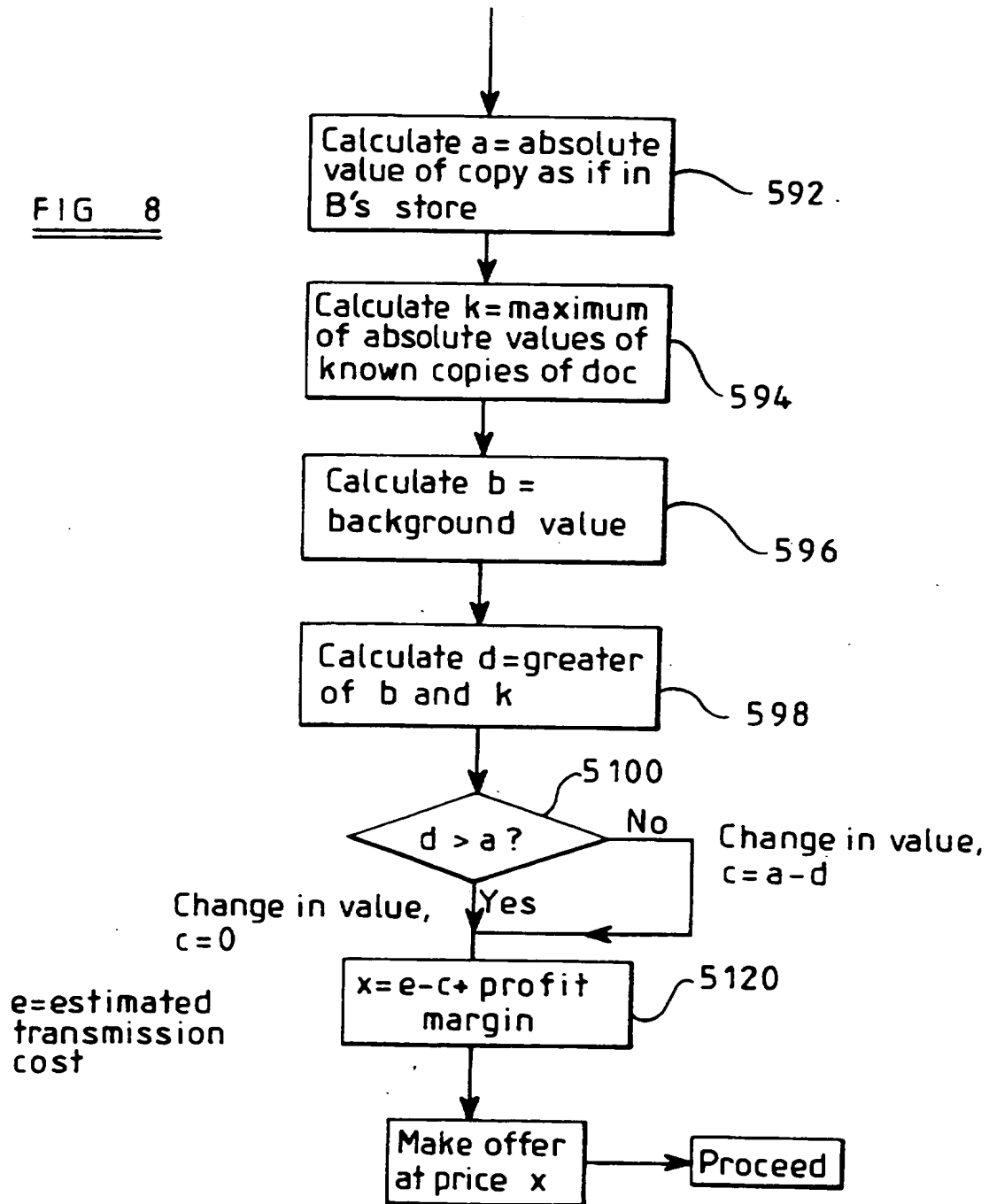
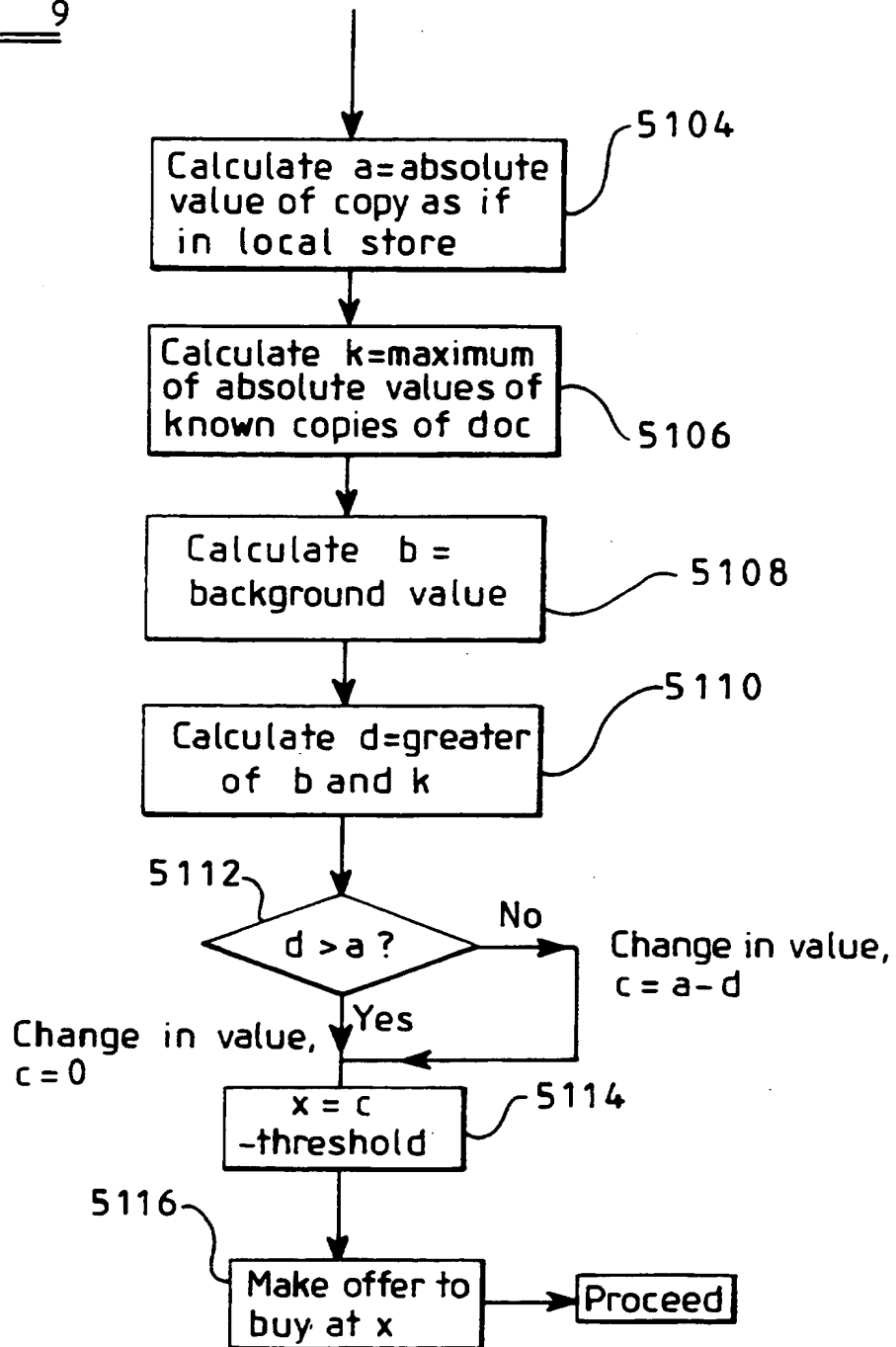
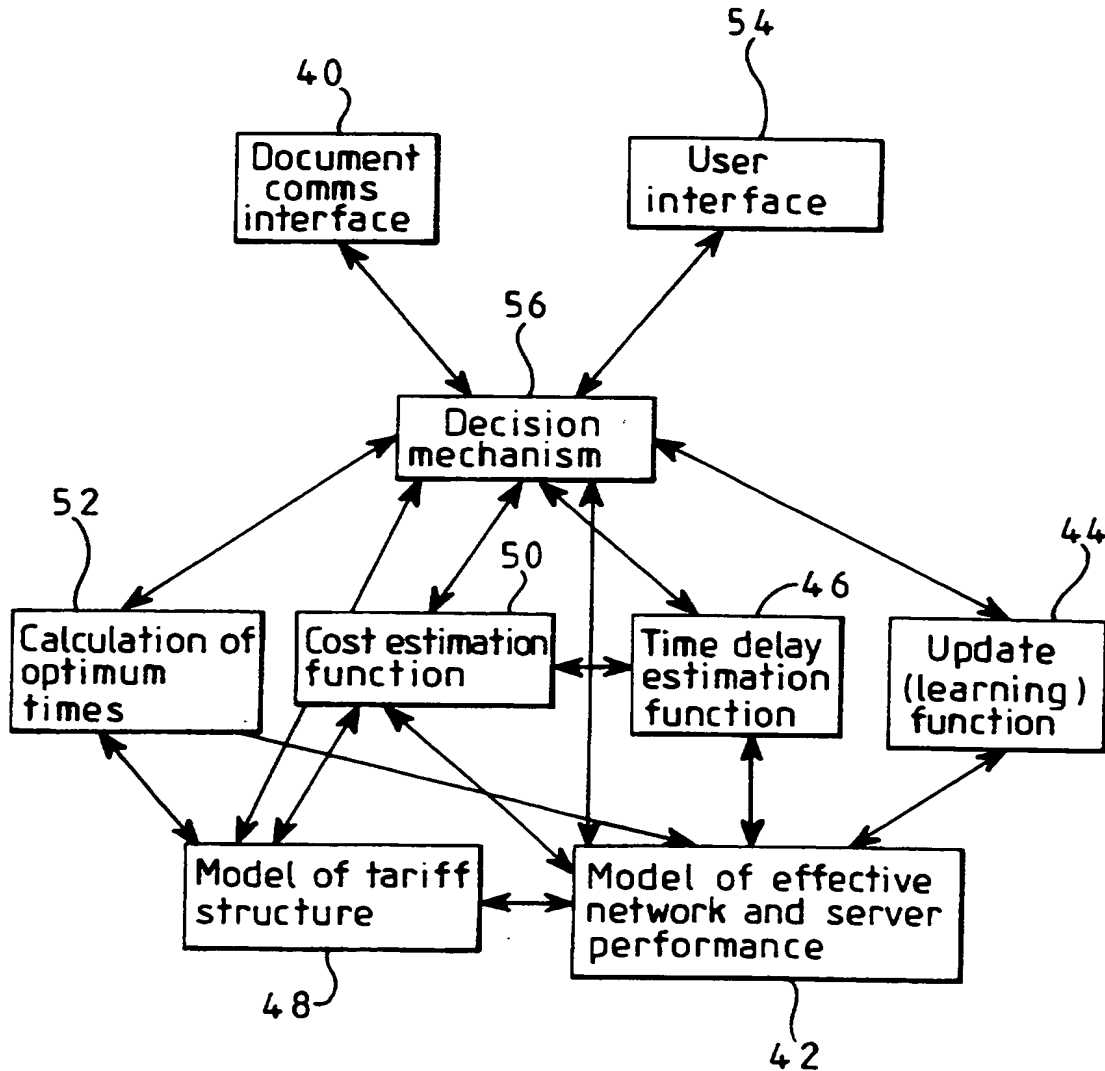
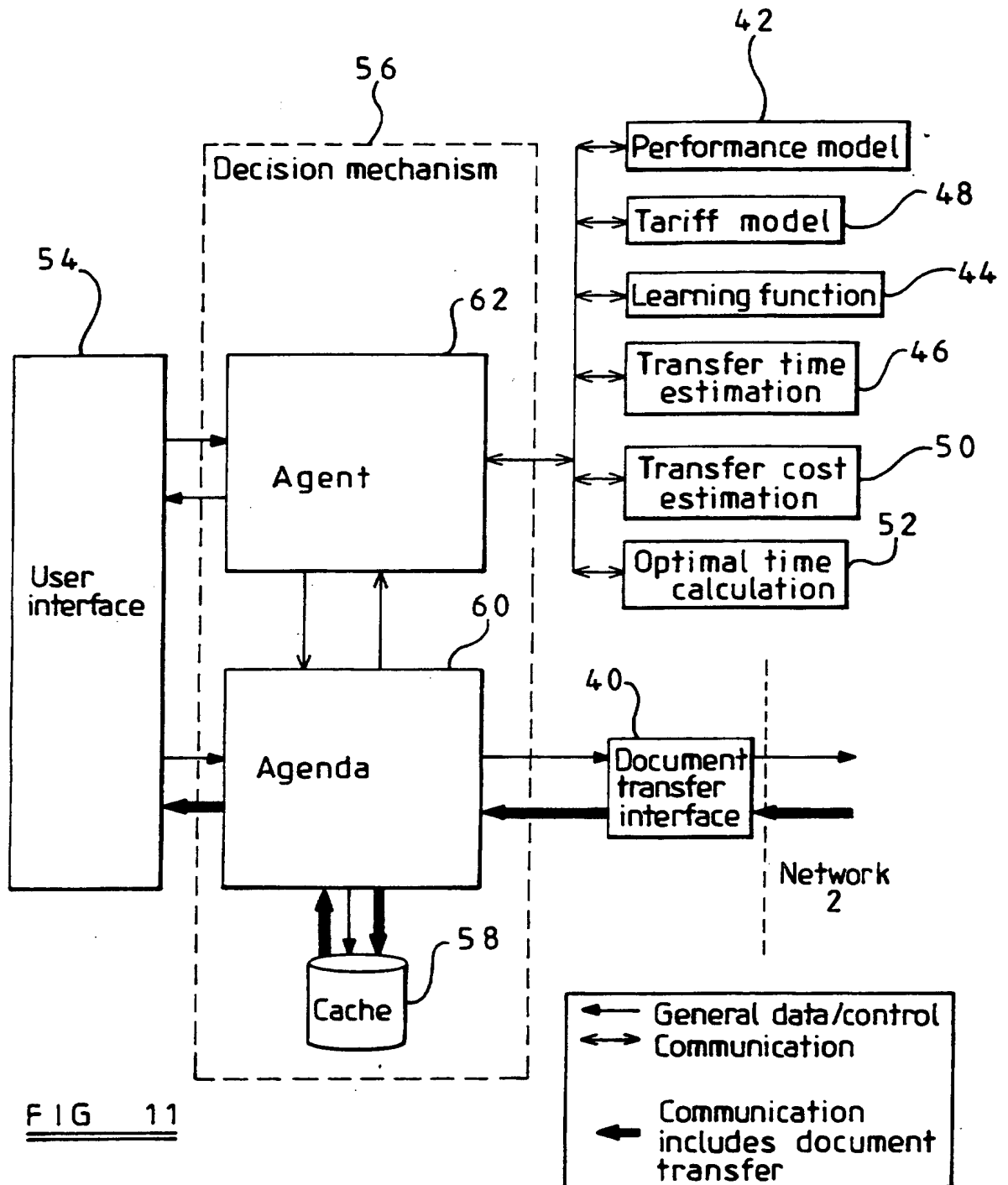
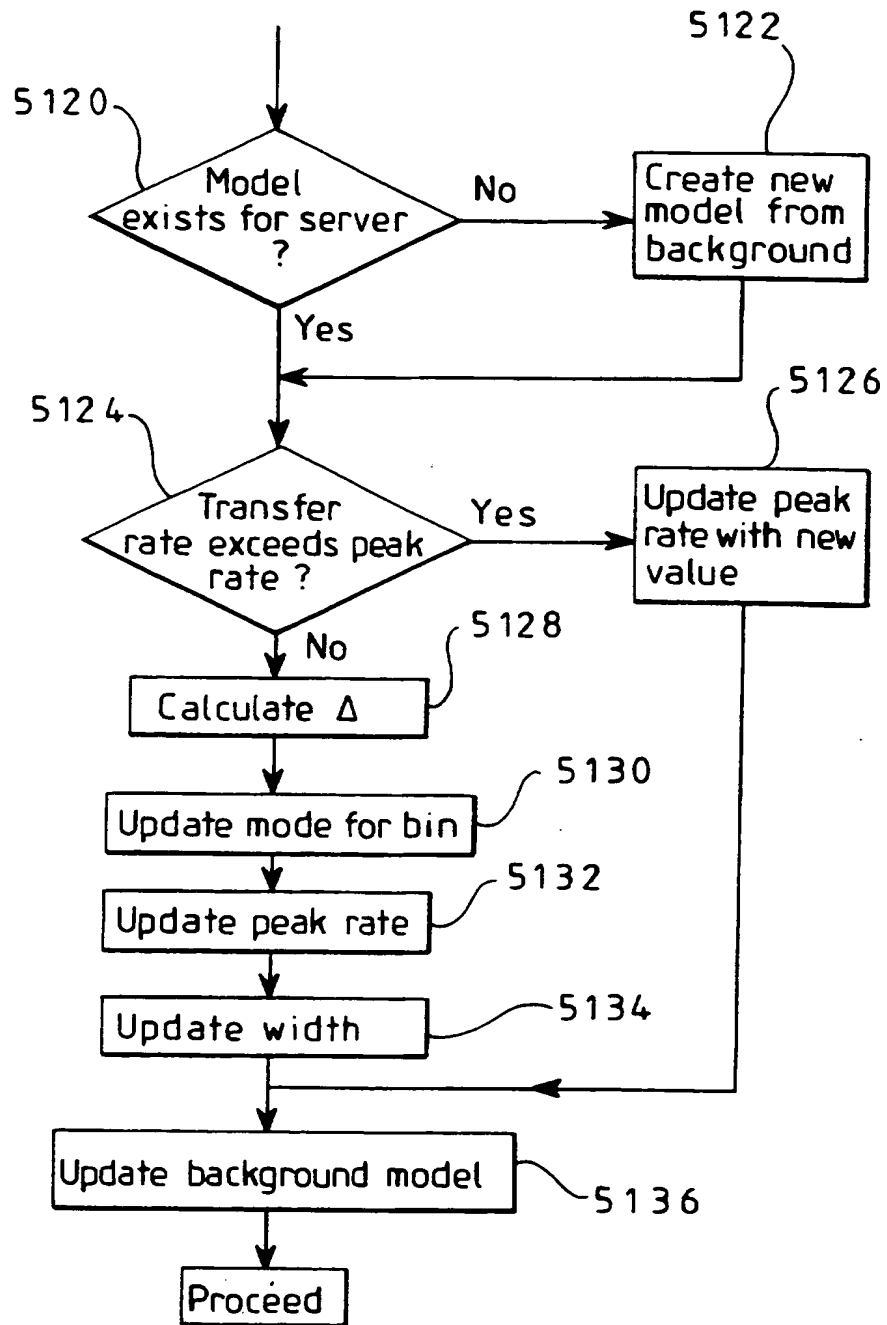
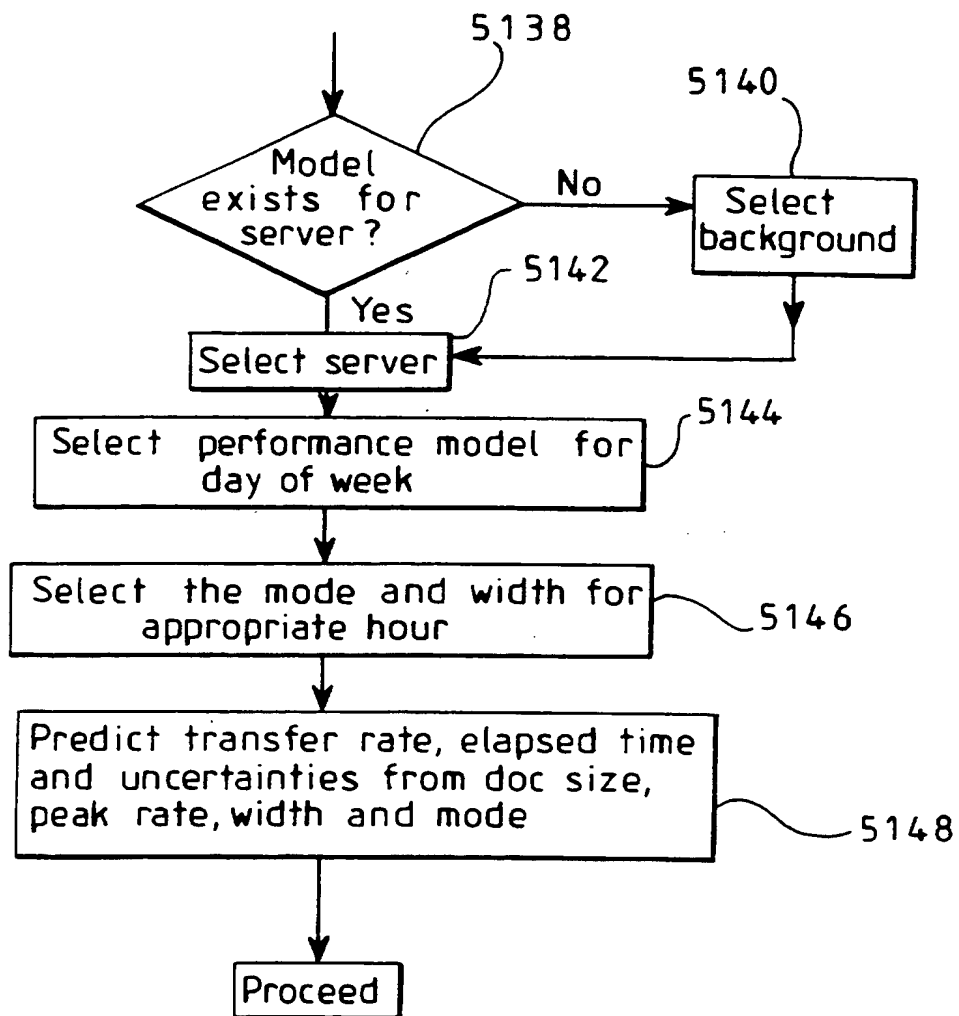
FIG 8

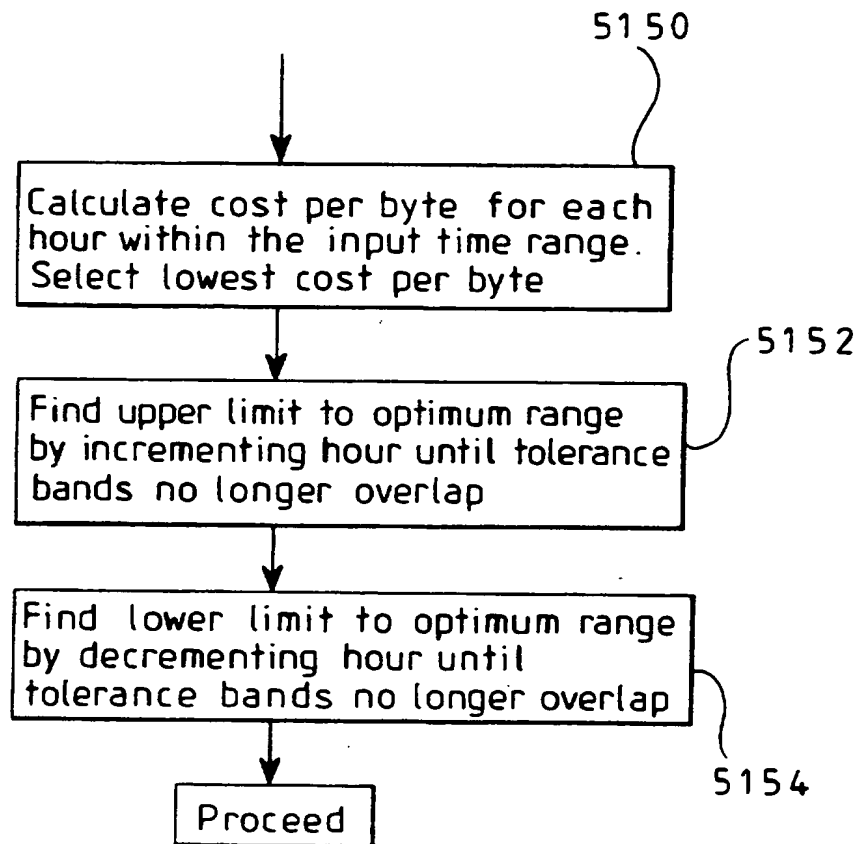
FIG 9

FIG 10



FIG 12

FIG 13

FIG 14

A DISTRIBUTED INFORMATION SYSTEM.

The present invention relates to a distributed information system, for example, the World-Wide Web.

A distributed information system (DIS) comprises a plurality of computers connected continuously or intermittently via a communications network. Each computer can comprise a server, a client, or a peer. A peer combines the functions of a client and a server. The server controls access to sources of electronic information, for example multimedia data, and can, in response to a request for information, transmit the information to a client on the same computer or to a client or a peer on another computer.

A client can request and receive information from the server on the same or another computer and can make the information available to other servers or peers or to a user via a user interface. The clients and peers can retain copies of the information (local copies) in order to avoid repeatedly transferring the information across the network. A storage area for local copies used for this purpose is known as a cache.

When information is received from the network, it is, for example, stored in the cache and passed to the user interface. Before the client sends a further request to the server for information, the client first checks whether the information is already held in the cache. If this is the case, the local copy of the information is passed to the user interface.

Otherwise, a request is made to the relevant server for the information.

The cache is of limited size and so, from time to time, the client deletes some of the contents of the cache to provide more space. Typically, the choice of what to delete is determined by how current the information is and how recently and/or frequently it has been requested.

Also, the problem exists of deciding whether to cache retrieved information and, if cached, whether the information is still equivalent to the original, which may have been updated.

An example of a DIS is the World-Wide Web (hereinafter referred to as the Web). The Web uses the Internet as an infrastructure. Clients and servers interact using a protocol called the Hyper-Text Transfer Protocol (HTTP). Servers can send copies of electronic information to clients in response to certain requests defined in the HTTP. The information can contain chains of references to other information (known as hyperlinks). A Uniform Resource Locator (URL) is used to identify the server and the name of the file within a given server's file space which contains the requested information.

Using the HTTP, it is possible for the server to tag transmitted information with details relating to expiry dates, no-cache warnings, and last modified dates in order to enable the client to determine the main issues mentioned above relating to caching.

Use of such a system can encounter difficulty in finding specific information, because there is an extremely large quantity of documents accessible via the Web. It is difficult therefore to discover which URL contains the most relevant information. Additionally, delays, the duration of which is difficult to predict, occur in retrieving an item of

information causing considerable inconvenience to the user. Because a principal variable cost involved is the charge for the telephone connection time to an organisation providing the internet connection service, such delays also incur cost penalties.

Delays can be reduced by providing caching. However, as described above, when a cache fills up, one or more items of information contained therein must be deleted. Consequently, the effectiveness of caching depends on the ability to rank documents such that information least likely to improve overall access efficiency is removed first.

A known Web client implements a local cache comprising a volatile cache in memory and permanent cache on a disc. A number of metrics and heuristics are used to determine whether information should be cached, for how long it will be consistent, i.e. its shelf life, and which information should be removed from the cache when it is full.

Caching can only improve access times for documents which have been requested at least once already. Products exist for performing prefetching i.e. block-transfer of documents in advance of the time at which a user needs to view them. The user must instruct such a system explicitly to perform a prefetch but is then free to perform other tasks until the whole block of information is ready to be viewed.

Web clients or "browsers" include tools to assist the user in recalling the URLs of useful documents and in organising such references according to subject. Such tools are called "book marks" or "hot lists". However, as in the case of "prefetchers", they rely on the user making decisions on which of the URLs to book mark and how to classify them.

In order to improve the efficiency and effectiveness of a DIS for a user without imposing demands in terms of specialist knowledge on the user, complex decisions and extra actions require software which is capable of making autonomous decisions on behalf of the user and to act upon such decisions. Software with such capabilities is known as an agent or, more specifically when forming part of a DIS, an information agent.

Known agents assist with tasks such as filing e-mail, information filtering and information discovery. Information filtering concerns automatic prioritisation or rejection of received information which is not of potential interest to the user. Information discovery concerns actively seeking information of potential interest which may be unknown to the user and may exist in remote servers. In order to perform such tasks, the agent must contain a model of the user interests and of the semantic content of the documents in question and must be able to calculate the degree of match therebetween.

A known agent architecture commonly comprises means for retrieving from the DIS information and other stimuli, means for responding to the stimuli, a model of the DIS and means for updating the model in response to new information becoming available in the DIS. A decision mechanism is also provided to determine what action the agents will take in response to a set of stimuli, the current state of the model and the state of the decision mechanism (which may include, for example, plans, intentions or commitments). The decision mechanism therefore selects those actions available so as, in the current circumstances, to lead to achievement of the agent's goals.

Known information agents deal with filing, filtering and/or discovering information without considering efficiency of retrieval. thus, a decision by a discovery agent to retrieve a document is based purely on how well the document matches the user interests whereas, given finite local storage, if the document can be retrieved quickly and cheaply, it may well be more efficient to make the user aware of the existence of the document but to leave the document where it is. Conversely, caching mechanisms address part of the retrieval efficiency problem without considering the document semantic content and how well it matches the user interests.

Web clients normally incorporate a user interface for displaying the information obtained from servers, in which case they are known as browsers. The usual way of finding information on a particular topic is to select the URL of a previously-known document containing information related to that sought, then to follow chains of references (hyperlinks) from this to other documents.

The only function in most known browsers which involves autonomous decision-making is control of the cache. The cache is local temporary storage space for documents imported over the network. Its purpose is to decrease response time for repeated requests to view the same document; after the first access, response is rapid as the document is held locally in the cache. When the cache fills up, one or more documents contained therein must be deleted. Typically in current browsers selected are those least frequently or least recently accessed. Storage and deletion of the cached documents is performed autonomously by the browser.

Known cache managers use relatively naïve functions to rank the cache contents for retention/deletion. The most sophisticated ones take into account the time taken originally to fetch the document, amongst other factors. This can be used as a crude prediction of the time and hence cost of re-acquiring the document. This is important for caching decisions, as generally it is sensible to cache documents which would take a long time to regenerate in preference to more easily accessible ones, given similar probabilities of repeated viewing requests. Many factors can affect transfer times, however, and so the original time taken is not necessarily a good predictor of the re-acquisition time. such factors introduce diurnal, weekly, and other variations.

Better predictions of transfer times and costs can also be directly of use to the user. For example, most browsers provide feedback to the user during document transfers on the likely time remaining until completion. This prediction is based on the actual transfer rate, the total size of the document (frequently known from the document header), and the amount of data transferred so far.

The normal mode of operation of the web and other hypermedia systems is for the user to request to view documents one by one. following a viewing request, the browser transfers and displays the document, which generally contains embedded references to other documents. The user requests to view one of these, which in turn is fetched and displayed, and so on. Pre-fetching involves the bulk transfer of several related documents in advance of specific requests to view them.

An existing software product which performs this function is Naminori Yarou from the Japanese software house BUG. The user runs this as a

separate application, selecting a root document, a maximum depth of links, and a time of day. At the given time of day, Naminori Yarou pre-fetches the "tree" of documents obtained by starting at the root document and following links to the specified depth. These documents are stored in a special directory in the computer and may be read using a general purpose web browser.

According to a first aspect of the invention, there is provided a distributed information system as defined in the appended Claim 1.

According to a second aspect of the invention, there is provided a distributed information system as defined in the appended Claim 20.

Preferred embodiments of the invention are defined in the other appended claims.

It is thus possible to provide a distributed information system which takes less time to access information and makes efficient use of disc space. Additionally, transmission costs are reduced and it is possible to speed up searches for information and to facilitate the pooling of information and computing resources among groups of users. Also, the network operator benefits from lower network congestion, improved quality of service and lower costs.

It is also possible to improve the efficiency and effectiveness of distributed information systems by means of improved predictions of elapsed times and costs of document transfers and of optimum times to perform such transfers. Possible benefits to a user include:

- 1) cost saving at the most economical times of day (this depends on both network tariffs and server and network performance); selecting among alternative (network) service providers; and selecting among alternative content sources (i.e. servers holding similar or identical documents),
- 2) time saving, i.e. reducing the time the user must devote to the process of transferring documents. The user can be warned in advance of slow transfers. Caching can be made more effective, thus increasing the probability that a requested document will already be in the cache. Prefetching, i.e. off-line bulk transfer of documents in anticipation of a user's requirements, is made more practical and more amenable to automation,
- 3) lower workload and knowledge required. The prediction method provides information which is needed for autonomous decision-making. This enables more functions to be performed automatically by software with less reference to the user. A simple example is that the need for the user to supply the time for a prefetch can be avoided and hence so can the need for the user to know when the most economical time is. The most economical time depends on effective server performance and on network tariffs.

It is further possible to provide an arrangement which learns about server and network performance and how it varies, for instance, at different times of day and from day-day. A model constructed by such a learning process can predict likely elapsed transfer times and in conjunction with knowledge of network tariff structure can predict likely cost. It can also be used to predict the most economical; time within given bounds to

perform a transfer. Thus, substantial savings in time and cost may be achieved.

It is further possible to provide an agent for a DIS which includes a decision mechanism for taking into account both intrinsic value and accessibility of a document and of other documents containing equivalent information. Such an agent may be used generically for a class of tasks which includes caching, filing, filtering and retrieval. These are performed in an integrated fashion and not as separate tasks. Such a DIS based on such a mechanism is robust, self-regulating and efficient and serves the interests of a network operator and also of individual users, groups of users and the user community generally.

The invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a block schematic diagram of a distributed information system constituting an embodiment of the invention;

Figure 2 is a block schematic diagram of an information trading agent of the system of Figure 1;

Figure 3 is a flow diagram illustrating a candidate transaction by the agent of Figure 2;

Figure 4 is a flow diagram illustrating an absolute value evaluation by the agent of Figure 2;

Figure 5 is a flow diagram illustrating a deletion of a cached document by the agent of Figure 2;

Figure 6 is a flow diagram illustrating an acceptance of a document by the agent of Figure 2;

Figure 7 is a flow diagram illustrating a request to sell a document by the agent of Figure 2;

Figure 8 is a flow diagram illustrating an offer to sell a document by the agent of Figure 2;

Figure 9 is a flow diagram illustrating an offer to buy a document by the agent of Figure 2;

Figure 10 is a block schematic diagram illustrating a time and cost prediction module of the system of Figure 1;

Figure 11 is a block schematic diagram of the layout of the module of Figure 10;

Figure 12 is a flow diagram illustrating a learning function of the module of Figure 10;

Figure 13 is a flow diagram illustrating a transfer time estimation of the module of Figure 10; and

Figure 14 is a flow diagram illustrating an optimum time range calculation by the module of Figure 10.

Like reference numerals refer to like parts in the drawings.

Referring to Figure 1, a distributed information system comprises a network 2, fixed computer equipment constituted by a plurality of workstations 4, mobile computer equipment constituted by a plurality of personal digital assistants (PDAs) 6, and an information service 8 having a database. Connections 10 are provided between the network 2 and the workstations 4 and the information service 8. Wireless links 12 are provided between the PDAs 6 and the network 2.

Each of the workstations 4, PDAs 6 and the information service 8 comprises an information trading agent 14, 16. As shown in Figure 2, each of the agents 14, 16 has a user interface 18, a message interface 20, a document interface 22, a cost evaluation module 24, a value estimation module 26, a modelling module 28 and a cache module 30, all of which are capable of communicating with a decision module 32. Additionally, the modelling module 28 can communicate with the cost estimation module 24 and the value estimation module 26, both of which can also communicate with each other.

The system constitutes a market for services - in general the agents 14, 16 are both providers of and customers for services. The agents 14, 16 buy and/or sell electronic information from and/or to each other in the form of multimedia documents.

The agent 16 for the information service 8 sends messages to the other agents 14, 16 informing them of the agent's 'resource competence' (a summary description of the subject matter held in its database), and of its 'root' document. The root document is a text document giving an

overview of services offered, and containing hyperlinks to other documents held in the database. Subsequently, the agent 16 behaves passively, i.e. the agent 16 responds to requests for information and purchase offers from other agents 14, but does not initiate any interactions.

An agent's competence is a measure of the overall quality of its document database. A competence record has the form *competence (Agent, Clist, Cmag)*, where *Clist* is very similar to a document content descriptor as described hereinafter, except that the numerical value associated with each topic may be greater than 1, and *Cmag* is the sum of the numerical values. *Clist* reflects the amount of information on the cited topics held in the database of *Agent*. It is calculated by summing the document content descriptors of all the documents in the database.

As well as responding in a similar manner to the agent 16, the agent 14 initiates interactions, both spontaneously and as a result of instructions from the user. Spontaneous behaviours include:

1. Periodically requesting acquaintances (with other agents) and details of their competencies;
2. Periodically selecting an acquaintance and informing it about a document in the database;
3. Periodically selecting a document, the possession of which is estimated to be of advantage to the agent's user, and making a purchase offer based on an estimated value; and
4. Deleting the least valuable documents in its possession whenever the database is overfull.

User-initiated behaviours include:

1. Making an offer to sell a document to another user;
2. Making an offer to buy a document from another user. The user request follows a search instruction, or else is initiated by the user clicking on a highlighted word when reading a text document containing a hyperlink; and
3. Searching based on a content profile, which is a set of keyword-number pairs describing the subject matter sought in the search and which is of the same form as the content part of a document record.

In order to deliver the documents, the agent 14, 16 must buy document transfer services from the network 2. In this example, a 'sender pays' convention similar to normal postal or telephone services has been adopted. However, it should be noted that the present invention is not limited to such a convention and others are envisaged. The agent 14, 16 selling the document must therefore take the cost of transmission into account when charging for the information service, i.e. when calculating an offer price.

A selling agent 14, 16 sets an offer price according to both an estimated value transaction and costs. Thus, if the transfer of the document gives a positive benefit to a user, the offer price may be below the transfer cost, for example, zero or a negative price.

Referring to Figure 3, negotiation of a candidate transaction between agents 14, 16 is initiated by either the provider of or customer for the information. This is done, for example, by the selling agent 14, 16 sending a message to a potential buying agent or information service 8 requesting that a document be bought (Step S2). A specific price may or may not be mentioned at this point. The agent 14, 16 receiving the offer

evaluates it in its own context (Step S4), and (after consulting the user if necessary) replies according to whether or not the offer is acceptable (Step S6). If certain of the terms of the transaction were not precisely specified (for example, the price), then the reply negotiates the terms which constrain acceptance (Step S8). For example, if the offer price is unspecified but the document is acceptable provided the price is at least £0.10 (Sterling), then the reply states this price, or perhaps a slightly higher one to allow scope for bargaining). If the reply is positive, the originator checks the terms (Step S10), confirms acceptance (if necessary) (Step S12), and the transaction proceeds (Step S14). If the reply is negative, then an improved offer may be made (Step S8).

As well as negotiating the purchase of services, the agents 14, 16 can exchange information, for example, on the existence, location, content, of a document sent unsolicited (as an 'advertisement', or to provide a tip to a colleague), or in reply to a request.

As an example of how the agents 14, 16 use this information when searching for information, the agent 14, 16 checks whether a document matching the given profile exists within the agent's own database. A check is then made as to the existence of such a document elsewhere. If neither of these checks yields a useful result, the agent 14, 16 checks whether it knows of any other agents 14, 16 which are likely to hold the document, and asks these agents for details of documents which match the profile best. If no such document exists, then the agent 14, 16 queries a directory agent (not shown) to establish the location of any suitable sources. Once a document has been identified and located, purchase negotiations proceed as described above.

In order to estimate the value of the candidate action, the agent generates a model of itself and its environment held as a collection of beliefs which are basically records associated with a time and which have the form *belief(Time, Document record)*.

The *document record* has fields pertinent to value calculations and comprises the location (i.e. in which agent's database the document is believed to be), a document content descriptor (see below), a unique document identifier, and the document size. Documents and copies are distinct in that several copies of the same document may exist (probably in different locations), with the same identifier, the content of these copies being identical.

A belief of the form *belief(Time, Document record)* means the agent believes that the document copy referred to was held in the specified location at *Time*.

The document content descriptor is a set of attribute-value pairs which describe the nature of the document content in terms of its degree of relevance to specified topics. The attribute component of the pair is a symbol denoting a topic; the value component is a number in the interval $[0, 1]$ corresponding to the relevance of the document to that topic (effectively it expresses membership of a fuzzy set denoted by the topic symbol). There is no restriction limiting the total relevance of a document summed over various subjects and it is quite possible for a document to have maximum relevance to several topics.

Normally, the content descriptor is constructed when a document is created, and will be copied without change for use in referring to that

document, except for the location fields of the descriptors referring to copies in different locations.

The universe of the fuzzy set is an arbitrarily large set of topics. If a topic is mentioned explicitly in the descriptor, then the value is degree of membership. Topics which are not mentioned explicitly in the descriptor have a membership value of zero.

For document content descriptors, there are basic set operations including:

1. A fuzzy membership operator which, given a descriptor and a topic, returns the degree of membership;
2. A number/length operator, which, given a descriptor, returns the sum of the degrees of membership;
3. A union operator, which, given (a set of) N descriptors, returns a descriptor in which degree of membership of a given element is the lesser of 1, and the sum of the memberships of the element in the input sets. Union of descriptors is interpreted as yielding the descriptor of a document obtained by concatenating the documents corresponding to the input descriptors; and
4. An intersection operator, which, given (a set of) N descriptors, returns a descriptor in which degree of membership of a given element is the product of the memberships of the element in the input sets. Typical use of the intersection operator is to determine how well the content of a document matches the description of an ideal document being sought.

Additionally, the following variants on the union operator which are used in dealing with collections (databases) of documents are defined as follows:

1. A sum operator in which the 'memberships' of the output are not capped at 1. Note that the output set cannot be interpreted as a fuzzy set and is not technically a document descriptor;
2. An average operator, which is a sum of N descriptors, followed by a division of all memberships by N. The result is a document descriptor, and is interpreted as the description of a typical document from within a set.

A variant on the intersection operator is also used and is defined as a 'dot-product' operator which intersects a set of document descriptors, and then finds the fuzzy length of the result.

Document goal records are used in calculating the value of a document distribution. They can be thought of as attractors which pull documents towards specific locations (i.e. agents' document databases). They have eight fields which divide into three groups. The first three fields describe document characteristics and a location from which it is desirable for such documents to be accessible. The next three are parameters in the calculation of a coefficient weighting the value of a given document's satisfaction of this goal. The last is a threshold below which contributions are ignored. The general form is:

document_goal_record (*Content profile, Location, Identity,*
Weight, Urgency, Remote utility,
Value threshold)

where:

Content profile has the same form as a document content descriptor, and is used to calculate how well a document's content matches this goal;

Location is the name of the agent 14, 16, from which it is desirable that certain documents are easily accessible;

Identity is a document identifier - certain goals apply to specific documents rather than documents matching a content profile;

Weight is a number indicating the importance of this goal record compared to others;

Urgency is a number indicating the importance of low access time for this goal;

Remote utility is a factor applied when a document location does not match *Location*. It is used to emphasise the importance of exact location as opposed to accessibility;

Value threshold corresponds to value contributions which below this threshold are ignored; and

There are two special document goal records which every agent has, and for which the location field is 'self', and the *Identity* is unspecified:

Interests goal record - the *Content profile* of this is the user's interests profile. The purpose of this goal is to ensure information of interest to the user is accessible from the agent;

Coherence goal record - the *Content profile* of this is the agent's competence profile. The purpose of this goal is to ensure that similar information tends to get stored together. The underlying supposition is that overall efficiency of information search and retrieval is enhanced by this measure.

In both these cases, the default *Weight* and *Urgency* values are 1 and the default thresholds are 0. The *Remote utility* defaults are 1 for the interests goal and 0 for the coherence goal.

Any number of other document goal records can be added. One use for these is to ensure that particular documents are not deleted. The document identifier is specified in the *Identity* field and the *Weight* parameter is used to indicate the importance of retaining the document.

An example of evaluating the absolute value of the document distribution is as follows. Considering the case of a single document contributing to a single goal, a first step is to calculate how well the document's contents match the goal. If the goal specifies a document identifier (Step S30), and this is the same as that of the document (Step S32), then using the dot product described above, the degree of match is $D \cdot D$ (Step S34), where D is the document's content descriptor. If the identifiers are different then the degree of match is 0 (Step S38). If the goal does not specify the identifier, then the degree of match is $G \cdot D$ (Step S36), where G is the goal's content profile. If the result is less than the value threshold (Step S40) then the document's contribution is ignored (Step S42).

This degree of match is then multiplied by a coefficient (Step S44) which is currently implemented as a step function multiplied by the goal's weight. The step function is 1 if the estimated delay in transferring the document to the agent in the goal's location field is less than 10 times the urgency of the goal, and otherwise 0. If the document and the goal's location do not coincide (Step S46), then the weight is further multiplied by the goal's remote utility factor (Step S48).

To obtain the document's value with respect to this goal, the result so far must be converted to money units (Step S50) (current conversion factor used is 10000 mp (millipence) per value unit) and then the estimated cost of moving the document to the goal location is subtracted. Additionally, local storage costs can be taken into account as well. In the single-document case, the full value is obtained by repeating the above calculation for each goal record, and summing the results.

In the case of multiple copies of a document held in different locations, the value of the most accessible (i.e. highest-scoring) copy of each document is counted to give the document's absolute value.

A more detailed description of this process now follows.

A document is at its most valuable when it is close to its likely point of consumption, i.e. it can reach the consumer in negligible time and at cost. If an agent thinks that its user will want to view a document in the near future, then it will attach a high value to holding that document locally. This evaluation is performed by comparing the document's content descriptor with similar structures representing the user's general

interests, any extant specific requests, and other significant factors. A first estimate is:

$$\text{Local value} = (A \sum_i S_i + BI + CL) \cdot D - \text{local costs} \quad (1)$$

where:

D is the document's content descriptor (obtained from the document's content record),

S_i are descriptors from any extant specific goals (the content profiles from general goal records). These represent transient requirements for documents on a given subject, or for specific documents.

I is the descriptor of the user interests (the content profile of the interests goal record)

L is a descriptor of the current database contents (the content profile of the coherence goal record) - this results in a tendency to specialise in a particular topic. It can be justified on the grounds of building up 'trading relationships' with other agents.

A, B and C are coefficients allowing differential weighting. Although these coefficients are independent of the document descriptors **D**, **S_i**, **I**, **L**, they are generally functions of other parameters as will be explained later.

Local costs take into account the value of space in the local database. The database has finite size so a cost is associated with occupying database hardware capacity based on the likely value of alternative occupancy. A first estimate assumes the cost to be zero if space is available above a threshold value and addition of a candidate document will not over-fill the database, and very high otherwise. The above

expression applies to a document stored close to its intended point of consumption.

A document held further away still has some value, but this is decreased due to the likely cost of transferring it plus any charge the proprietor of the database might make, and also due to the increased response time. If a document has been located on a remote database and the cost of access and transfer and the 'guaranteed' maximum delivery delay are obtainable from the other agent and the network, the remote value is simply the local value (without the term in L) minus the costs, if time is no problem. The significance of delay time depends on how urgently the information is needed (if a specific request has been made) or on likely useful response time if a user requirement is being anticipated. One way of handling this is to make A and B monotonically decreasing functions of delay time. A is also a function of temporal requirements specific to a query. Delay times themselves are likely to be functions of service level, network and time of day. Thus:

$$\text{Remote value} = (\sum_i A_i(\Delta) S_i + B(\Delta) I) \cdot D - \text{costs} \quad (2)$$

where:

Δ is the delay time

$A_i(0) = A$ in (1)

$B(0) = B$ in (1)

The above equations assume that the local database is managed by the evaluating agent and that the intended point of consumption is close to that database. This is appropriate for evaluating whether to request transmission of a remote document to the local database or leave it

where it is. The equations are special cases of a more generally-applicable absolute value function:

$$\begin{aligned} \text{Absolute value} = & \sum_i \max_{loc} (\Phi_i(A_i(\Delta_i)S_i.D, \text{cost}_i)) \\ & + \max_{loc} (\Phi_b(B(\Delta_b)I.D, \text{cost}_b)) \\ & + \Phi_i(CL.D, \text{local cost}) \end{aligned} \quad (3)$$

where now:

S_i are descriptors from any specific document movement goals (not just directed towards the local database),

Δ_i and cost_i are the cost and delay time associated with movement goal i ,

Δ_b and cost_b are the cost and delay time associated with movement of the document from its current location to a process which can display the document to the user,

$\Phi_i(x, y)$ is more complicated than simple subtraction, as a negative component of value from one goal should not detract from positive values from the others. An example of the function is $\Phi_i(x, y) = x - y$ if $x - y > z(i)$, and 0 otherwise. $z(i)$ is simply a threshold ≥ 0 which may or may not depend on i ,

$\max_{loc}(X)$ selects the greatest X over values corresponding to copies of the document(s) in different locations,

L is the empty fuzzy set (i.e. $L.D = 0$) if the document location in question is not managed by the evaluating agent

In order to evaluate changes in the distribution of information, the effect on the value of the candidate action is estimated. Thus the value increments resulting from adding a new copy at a specified location and/or deleting a copy from a specified location is used.

Three values are used to calculate the value increment described above. These are:

- a) the absolute value of the copy in its current location (when deleting) or proposed location (when adding);
- b) the absolute value of the highest ranking known alternative copy of this document;
- c) the background value (see below) of the document distribution, i.e. an estimate of the accessibility of the information in this document from other sources.

The background value is estimated by selecting the known agent whose competence best matches the content of the document in question, and performing an absolute value calculation as if a copy were present there, weighted by a factor to allow for the probability that information equivalent to the document's contents is indeed present in that agent's database.

The difference between a) and the higher of b) and c) is termed the incremental value of a copy. The value is used in deciding whether to delete a document. The difference between the document's value and that of a hypothetical copy in another location is termed its relative value. The relative value is used in deciding on the merits of document movement.

The agent performs its own cost and time estimation based on data obtained actual document transfers. This estimation function takes as arguments: *Size*, the size of document to be transferred (obtained from the document record), and *Agent 1* and *Agent2*, the identities of source

and destination agents. Currently, the returned values, *Delay* and *Cost*, are calculated using simple functions of *Size* and the transfer index for the agent pair. This index is recalculated from cost and time figures each time a new service is negotiated with the network, so the estimation is reasonably accurate if document exchange between the agents is frequent.

Once a change in the value has been determined, a decision is made regarding whether to proceed with the candidate action based on comparing the estimated change in value of the document distribution associated with taking the action with the estimated cost.

In the case of adding a copy to the agent's own database, the charge is paid to the agent owning the copy of the document from which the new copy is to be made. Since, in the present example, the sender pays for document transfer, this will typically include recoupment of the transmission cost. In the case of sending (selling) a copy to another agent, the cost is the transmission cost minus the charge paid by the other agent. In the case of deleting a copy, the cost is zero. Thus, the decision is taken to proceed if $\text{change in value} - \text{cost} > \text{threshold value}$.

If there are several candidate actions, and more than one satisfies the criterion, these can be ranked in accordance with $\text{change in value} - \text{cost}$, and the first n chosen for execution.

The following examples are evaluations of candidate actions.

Referring to Figure 5, when the contents of its document store exceed a certain size (Step S52), the agent 14, 16 selects a document for deletion. Since for deletion, the cost is zero, this is done purely on the basis of

change in value. For each document in the cache, the following steps are performed:

- 1) the copy's absolute value (a) is calculated (Step S54);
- 2) the absolute values of all other known copies are calculated and the greatest value selected (the best known alternative, k) (Step S56);
- 3) the background value is calculated (b) (Step S58);
- 4) the greater of b and k is selected as the replacement value (r) (Step S60);
- 5) if $r > a$ (Step S62), then the change in value is zero, otherwise, the (negative) change in value is $r-a$.

The copies whose change in value have the greatest (i.e. least negative) values are deleted (Step S64) until the store size threshold is no longer exceeded.

If a first agent receives a request from a second agent that the first agent buys a document from the second agent for x amount of money, this covers a spectrum of scenarios including:

1. mail: the second agent makes no charge for the document which means it is paying for delivery just like when one makes a phone call or posts a letter;
2. promotion: the second agent makes a negative charge, effectively paying the first agent to take the document to encourage the first agent's user to read e.g. some promotional material,
3. sale (Figure 6): a straightforward commercial offer in which the second agent bases its offer on the value of the document content plus transfer cost plus profit.

The first agent therefore evaluates the offer in the following way:

- 1) The agent calculates the absolute value a copy would have if it were located in its cache (a) (Step S66);
- 2) the best known alternative value (k) is calculated as above (Step S68);
- 3) the background value is calculated (b) (Step S70);
- 4) the greater of b and k is selected as the absolute value of the document (d) (Step S72);
- 5) if $d > a$ (Step S74), then the change in value (c) is zero, otherwise, the change in value is $a - d$;
- 6) if c exceeds the cost of the transaction by more than the required amount ($c - x$ threshold) (Step S76), then the first agent decides to accept the offer.

If the first agent receives a request from the second agent that the first agent sells a document to the second agent for x amount of money (Figure 7), this receiver-initiated scenario is a typical client-server scenario with variations according to how much the first agent perceives it as being in its own interests that the second agent receives the document.

The first agent therefore evaluates the offer in the following way:

- 1) The first agent calculates the absolute value (a) a copy would have if it were located in the second agent's cache (Step S78). This may possibly be greater than the value of the first agent's own local copy if the first agent has a goal for the second agent to possess the copy, but will usually be lower;
- 2) the best known alternative value (k) is calculated as above (Step S80). This will normally be the first agent's own local copy;

- 3) the background value (b) is calculated (Step S82);
- 4) the greater of b and k is selected as the absolute value (d) of the document (Step S84). This will normally be the value of the first agent's own local copy.
- 5) if $d > a$ (Step S86), then the change in value (c) is zero, otherwise, the change in value is $a - d$
- 6) if x plus the change in value (c) exceeds the estimated cost of the transmission (e) b more than the required amount ($x + c - e >$ threshold) (Step S88), then the first agent decides to accept the offer (s Step S90).

Part of the role of the agent 14 is spontaneously to offer documents for sale to other agents. This can occur in scenarios such as:

1. Commercial sale: the first agent has a valuable document which it wants to exploit commercially;
2. Promotion: the first agent wants other users to read the document and so is prepared to offer it a 0 or even negative cost;
3. Information sharing (for example, within a work-group): the first agent is part of a group of agents whose users have agreed to pool information. When the first agent acquires a document which it thinks another agent would find of interest, it offers to sell it to this agent. Although the agents are collaborating, the arrangement is kept on a commercial basis so that costs are shared equably.

If the first agent selects a document in its database, the second agent by comparing the document's content descriptor with the first agent's competence, can then formulate an offer in the following way:

- 1) the first agent calculates the absolute value (a) a copy would have if it were located in the second agent's cache (Step S92). This may possibly be greater than the value of the first agent's own local copy if

the first agent has a goal for the second agent to possess the copy, but will usually be lower;

- 2) the best known alternative value (k) is calculated as above (Step S94). This will normally be the first agent's own local copy;
- 3) the background value (b) is calculated (Step S96);
- 4) the greater of b and k is selected (Step S98) as the absolute value of the document (d) . This is normally the value of the first agent's own local copy;
- 5) if $d > a$ (Step S100), then the change in value (c) is zero, otherwise, the change in value is $a - d$;
- 6) the first agent sets the offer price (x) as $x = e - c + \text{profit margin}$ (Step S102) where e is the estimated cost of the transmission.

Part of the role of the agent is to autonomously acquire documents from other agents if it considers it cost-effective to hold the documents locally. Suppose the first agent learns by some means (for example, via a message from another agent containing the document descriptor) that the second agent possesses a document of interest. The first agent formulates an offer in the following way (Figure 9):

- 1) The first agent calculates the absolute value (a) a copy would have if it were located in the first agent's cache (Step S104);
- 2) the best known alternative value (k) is calculated as above (Step S106);
- 3) the background value (b) is calculated (Step S108);
- 4) the greater of b and k is selected as the absolute value (d) of the document (Step S110);
- 5) if $d > a$ (Step S112), then the change in value (c) is zero, otherwise, the change in value is $a - d$;

- 6) A sets the offer price (x) as $x = C - \text{threshold}$ (Step S114), and makes an offer (step 116).

Each of the work stations 4, PDAs 6 and the information service 8 further comprises a time and cost prediction module as illustrated in Figure 10. The module comprises a document communications interface 40 for connection to the network 2 for receiving "documents" or information. A model of effective network and server performance 42 represents partial knowledge about patterns of effective performance of the network as a whole and in respect of document transfer from specific servers. An update or learning function 44 updates the model 42 on the basis of measurements derived from actual document transfers. A time delay estimation function 46 estimates the time delay associated with a document transfer at a given time based on the model 42. A model of tariff structure 48 represents the network tariff structure so as to enable estimated transfer times to be expressed as costs. A cost estimation function 50 estimates transfer costs based on the tariff structure model 48, the performance model 42 and the function 46. A calculation 52 of optimum times uses the performance model 42 and the tariff structure 48 to estimate optimum times to perform transfers within the distributed information system. A user interface 54 allows interaction with a user including displaying documents requested by the user. A decision mechanism 56 decides upon and initiates appropriate actions, such as user interface actions, cash operations and sending or accepting documents. The decision mechanism 56 may act autonomously or in response to external stimuli such as user interface events, elapsed and absolute time, and cache state taking into account the time and cost functions at 46, 50 and 52.

When used in a client or peer as part of the Web, an Intelligent Web Browser (IWB) is provided having novel and advantageous features compared with known browsers. The layout of the module of Figure 10 is shown in more detail in Figure 11 and each of the sub-modules will now be described.

The user interface 54 comprises a conventional or known Web Browser. Communication between the user interface 54 and the decision mechanism 56 is substantially analogous to that between a known browser and a known proxy server. Thus, the interface 54 sends requests for a document described by a URL and the reply from the decision mechanism 56 includes the requested document and/or other relevant information. The reply may be generated by the mechanism 56 itself or may be obtained across the network 2 from another source.

The document transfer interface 40 permits client-server communication across the network 2 to obtain documents and other information. For instance, standard Internet protocols such as HTTP may be used.

The decision mechanism 56 comprises a cache 58, an agenda 60 and an agent 62. The cache provides persistent local storage of documents or copies of documents under control of the agenda 60. The cache 58 makes use of the file system of the local computer.

The agenda performs functions relating to direct management. For instance, the agenda 60 controls storage in, deletion from, and retrieval of documents from the cache 58. The agenda also manages the obtaining of documents across the network 2 by the document transfer interface 40. The agenda further manages execution of prefetch

instructions as will be described hereinafter. Further, the agenda 60 responds to requests from the user interface 54 using standard protocol identifiers.

The agent 62 provides a more abstract and indirect level of management for the IWB than the agenda 60 and also stores information used in such management decisions. In performing such functions, the agent 62 makes extensive use of information supplied by the estimation and modelling functions 42, 46, 48, 50 and 52. For instance, the agent 62 calculates cache retention priority for documents based on information deduced from the document, information supplied by the user or deduced from user behaviour patterns and information supplied by the estimation and modelling functions.

The agent 62 responds to prefetch requests from the user by constructing more explicit prefetch requests and supplying them to the agenda 60. Tasks performed in constructing such requests include determining the optimum time to begin prefetching (possibly within constraints supplied by the user) and generating repeated requests for documents to be prefetched on a regular basis. The agent 62 also responds to other requests for information or action from the user via the interface 54.

The agent 62 interacts with the agenda 60 to influence the operation of the agenda and to obtain information, for instance about the cache and about performance measurements of document transfers which have taken place over the network 2. The agent 62 manages its own persistent information stores and oversees those of the performance and tariff models 42 and 48. This management includes occasional purging of information which the agent 62 judges to be less important to retain.

The performance model 42 comprises an arbitrary number of performance models of respective individual servers connected to the network 2 and a performance model of an average or "background" server. The performance model of each server is identified by the name of the server. Overall performance capacity of the server is represented by a single number in units of bytes per second and known as the server peak rate.

Each model is divided into three separate models describing diurnal variations in performance for week days, Saturday and Sunday. The reason for this further division is that network traffic patterns for working days are generally different from those for weekends and many sites exhibit anomalous performance patterns at specific times over weekends, possibly because of routine maintenance tasks. Each diurnal model is composed of twenty four bins providing one bin per hour of the day. Each hour is represented by two numbers representing mode and width. The mode is the predicted average transfer rate during that hour expressed as a fraction of the peak rate. The width is the tolerance of the mode and represents a measure of the variability of the transfer rate expressed as a fraction of the peak rate.

The tariff model 48 contains one or more alternative tariffs with one of these being nominated as the currently applicable tariff. The tariff model 48 represents, for instance, the tariff structure of a telephone company. The model described herein does not allow for varying distance of call (although it could be generalised to do so) as it is intended to be applicable to calls between a fixed location, such as the user home or office, and a local Internet access provider.

The tariff model is made up of an arbitrary number of tariff bands, each of which is defined by the following parameters:

units i.e. the charge per unit time for a call within this band;
 time-band. low i.e. the hour in the day at which this band starts;
 time-band. high i.e. the hour in the day at which this band ends;
 day-band. low i.e. the day in the week at which this band starts; and
 day-band. high i.e. the day in the week at which this band ends.

The learning function 44 takes a single measurement of effective transfer rate in bytes per second from a given server at a given time and uses it to update the performance model 42. During normal use, this is performed after completion of a document transfer. During a "training" mode, it is applied repeatedly using recorded data.

The operation of the learning function 44 is illustrated in Figure 12. Step S120 checks whether a model exists for the server in question and, if not, a new model is created at Step S122 by replicating the background model. The particular server model is then updated as follows.

At step 124, the actual transfer rate is compared with the peak rate and, if greater, is replaced by the new value at Step S126. Otherwise, the difference between the actual value and the predicted value for this time is calculated and divided by the peak rate to give Δ at Step S128. The predicted value is the peak rate multiplied by the mode parameter for the appropriate bin. At Step S130, the mode for the bin is updated according to:

$$\text{mode}' = \text{mode} + \Delta * \sigma_m / \text{width}$$

where σ_m is a learning parameter, for instance of the order of 0.05. If the result is less than zero, the new mode is set to zero. If the result is greater than one, the new mode is set to one.

In Step S132, the peak rate is updated according to:

$$\text{peak rate}' = \text{peak rate} * (1 + \Delta * \sigma_p / \text{width})$$

where σ_p is a learning parameter of the order of 0.001. At Step S134, the width for the bin is updated according to:

$$\text{width}' = \text{width} + (|\Delta| - \text{width}) * \sigma_w$$

where σ_w is a learning parameter of the order of 0.2. The width value is capped at upper and lower limits, for instance of 0.7 and 0.03.

The background model is updated at Step S136 in the same way except that the background peak rate is first set equal to the peak rate of the current server. The modes and widths of the background model therefore express an average server performance model relative to peak rate but the background peak rate is taken to be the peak rate of the most recently accessed server.

The transfer time estimation 46 estimates transfer time from the performance model 42, a time including day of the week, a document size in bytes, and the server name. This is illustrated in Figure 13. Step S138 checks whether a model exists for the particular server and, if so, selects it at S142. Otherwise, the background model is selected at S140. Step S144 selects the performance model for the appropriate day of the week and step S146 selects the mode and width of the performance model corresponding to the appropriate hour. Step S148 then predicts the transfer rate, elapsed time and uncertainties from the document size, peak rate, width and mode. In particular, the transfer rate is:

peak rate * mode

with uncertainty:

peak rate * width.

The predicted elapsed time transfer is:

peak rate * mode * document size

with uncertainty:

peak rate * width * document size.

The transfer cost estimation 50 calculates a transfer cost estimate from the tariff model 48, the transfer time estimation 46, the time and day of the week, the document size in bytes and the server name. In particular, the charging rate, for instance in pence per second, is obtained from the tariff band appropriate to the time of day and day of the week. The predicted transfer rate and uncertainty are calculated as described hereinbefore. The predicted cost per byte is then:

charging rate/transfer rate

with uncertainty:

charging rate * transfer rate uncertainty/(transfer rate)²

The predicted cost for transfer is:

cost per byte * document size

with uncertainty:

cost per byte uncertainty * document size.

As shown in Figure 14, the optimum time calculation 52 calculates the optimum time range for performing a transfer within a given time range from the server name and the models and functions described hereinbefore. At step S150, the cost per byte for each hour within the input time range is calculated and the hour corresponding to the lowest

cost per byte is selected. The input time range is normally the time period from the present time to the time which is specified by the user and which indicates the time by which the document must have arrived. It is preferable to provide an optimum time range for transfer rather than a specific time because the software may be prevented from commencing transfer at exactly the planned time, large documents may take a long time to transfer, and transfer may be interrupted and may have to be repeated. The upper limit to this range is obtained in step S152, starting from the best hour, by incrementing the hour until:

$$\text{cost per byte} - \text{best cost per byte} > (\text{uncertainty on cost per byte} + \text{uncertainty on best cost per byte})/2.$$

The lower limit is determined in step S154 by starting from the best hour and decrementing the hour until:

$$\text{cost per byte} - \text{best cost per byte} > (\text{uncertainty on cost per byte} + \text{uncertainty on best cost per byte})/2.$$

In order to illustrate operation of the module, various specific examples will be described hereinafter.

Warning of expensive document transfers

In the IWB, the user may request to be informed if a requested document transfer is estimated to exceed a user-supplied figure. The user may make this request via a dialogue box displayed by the user interface 54. The user interface 54 notifies the agent 62 that this so-called thresholding function is to be enabled and of the threshold figure.

When the agenda 60 receives a request to fetch a document which is not in the cache 58, it notifies the agent 62 of the request and awaits

permission to proceed. If thresholding is enabled, the agent 62 calculates the transfer cost estimate as described hereinbefore. If the estimate exceeds the threshold, then the agent 62 causes the user interface 54 to display a dialogue box notifying the user of the estimated cost and asking whether to proceed. The dialogue box contains buttons allowing the user to answer 'yes' or 'no'. If the user indicates 'no', then the agent 62 provides the agenda 60 with a document containing a confirmation message to pass to the user interface 54 in lieu of the requested document. If the user answers 'yes' or the threshold is not exceeded, the agent 62 tells the agenda 60 to proceed with the transfer.

The estimation function requires a document size to perform its calculation. The agent 62 maintains a record of known documents including their sizes. If the size of the document is known from this record, then this figure is used. If the size of the document is not known, the IWB uses a default document size (dependant on the media type). There are alternative means of discovering the size, however, such as requesting the server to send the header of the document only. Typically, this contains the required size information.

Suppose the document size is 200 KB, the current time is 14:30 on a Wednesday, and the user has set a threshold of 10Op. Using typical server and tariff models gives:

charge rate = 0.3p/s

mode = 0.0442142, width = 0.129002) Peak_rate = 4293.41

transfer rate estimate = $4293.41 * 0.0442142 = 189.8$ byte/s (very low)
 with uncertainty $4293.41 * 0.129002 = 553.8$ byte/s

transfer cost estimate is $200000 * 0.3 / 189.8 = 316p$

This is well above the threshold and the user is asked for permission to proceed.

Prefetching

On request from the user, the user interface 54 displays a prefetch request form for a tree of documents originating at a given root document. This form allows the user to select various options and specify various pieces of information to tailor the request. Mostly these default to sensible values and so the user often only has to make one or two choices and click a button to submit the form. For the purposes of this example, the important pieces of information to be specified on the form are:

- 1) the time by which the user wants the document available. This will be the earliest time that the user is likely to need to use it. In the case of a daily newspaper, this time may be, say, 8:00 on the day of issue. The default is 'as soon as possible'.
- 2) the time after which the document will no longer be useful. In the case of a daily newspaper, this time may be, say, 20:00 on the day of issue. This defaults to 1 day after the 'prefetch by' time.

On submission, the information from the form is passed to the agent 62. The agent 62 must then translate the information into more definite terms, and pass it on to the agenda 60 for execution of the request. For the purposes of this example, the main task the agent 62 performs is to convert user-supplied times to a pair of times indicating when the agenda should commence attempting to perform the prefetch and when the agenda should abandon attempts to perform the prefetch. If an attempt to prefetch is unsuccessful, the agenda 60 will retry at intervals until the 'abandon prefetch' time is reached.

To perform this task, the agent 62 makes use of the optimum time calculation 52 which takes a time range as input. As the lower bound of this range, the agent uses current time or the document publication time if this is known and is later than the current time. For the upper bound, the user-supplied 'prefetch by' time is used. The optimum time calculation 52 returns a time range within the range supplied by the agent during which document transfers from the site holding the root document are estimated to be most economical. The agent 62 uses the lower bound of this range as the lower bound of the range to pass to the agenda 60 and the 'no longer useful' time as the upper bound. This means that prefetch attempts will start at the beginning of the most economical range. It is likely that a successful transfer will be achieved by the end of the most economical range but, if necessary, attempts will continue until the document is no longer useful.

The agenda 60 holds the prefetched documents in the cache 58. The agent 62 can request information regarding cache contents from the agenda 62 including whether a cached document was transferred as part of a prefetch request. Once the 'abandon prefetch' time is exceeded, the

prefetched documents are no longer marked as such. Part of the agent's role is to calculate a 'cache retention priority' value for each cached document. Documents which have been prefetched are allocated a high retention priority and so the agenda 60 will retain them in the cache until the 'no longer useful' time at least.

Suppose the current time is 19:00, the user-supplied 'prefetch by' time is 08:00 tomorrow, the publication time is 03:30 daily, and the user-supplied 'no longer useful' time is 17:00 tomorrow. These times are appropriate for a scenario in which the user wishes to read an electronic newspaper on a portable PC on the way to work without using an expensive cellular telephone account.

The times used as input to the optimum time calculation 52 are:

lower bound = next publication time = 03:30 tomorrow

upper bound = 'prefetch by' time = 08:00 tomorrow

The whole of this time period falls within a 'low charge' band of the tariff. Consequently, the optimum transfer time corresponds to the highest transfer rate, i.e., using typical performance and tariff models:

time bin 5 (first bin is bin 0), time = 5:00 - 5:59

transfer rate = peak rate * mode = $4293.41 * 0.649703 = 2789.4$ bytes/s
 with uncertainty = peak rate * width = $4293.41 * 0.240578 = 1032.9$ bytes/s

cost per byte = charge rate/transfer rate = $0.1/2789.4 = 0.0000968$ p/byte

with uncertainty = charge rate * transfer rate uncertainty
 $(\text{transfer rate})^2 = 0.1 * 1032.9 / (2789.4)^2 = 0.0000133$ p/byte

The agent 62 expands around the optimum time to obtain the optimum range as described hereinbefore. The optimum time range may be calculated as running from bin 5 to bin 7, i.e. 5:00 until 8:00. The agent 62 therefore instructs the agenda 60 to start prefetching at 5:00 and to terminate attempts at 17:00.

Cache retention priority

As mentioned above, part of the role of the agent 62 is to calculate a cache retention priority (CRP) parameter which the agenda 60 uses to order the documents in the cache for deletion.

The agent takes into account the cost of retrieving the document in this calculation. All other factors being equal, a document which is expensive to retrieve should be more likely to be retained (high CRP). Clearly, other factors need to be taken into account as well, e.g., likelihood that a document will be needed again (CRP increases with re-access probability), and disk space occupied by the document (CRP decreases with disk space occupancy). The user's explicit wishes and those implicit in prefetch instructions also need to be taken into account.

The agent 62 calculates CRP as a number between 0 and 1 as follows. While this is not intended as a definitive formula, it serves as an

illustration of how the estimation functions and performance models are very relevant to cache management:

- 1) If the user has explicitly indicated a CRP value, this value is used. (The user interface 54 provides a facility whereby the user may do this)
- 2) If the document has been prefetched and the current time is before the 'no longer useful' time, then the $CRP = 1$.
- 3) Otherwise, the CRP is calculated as:

$$(1 - \exp(-P \cdot \theta(t_c))) / (1 + \exp(-P \cdot \theta(t_c)))$$

where P is a measure of how frequently the document is likely to be accessed by the user, and $\theta(t)$ is dependant on predicted transfer rate as follows (t = time, t_c is the current time):

$$\theta = K * \text{charge rate} / (\text{transfer rate})^2 * \text{size in bytes} / \text{storage charge/delay tolerance}$$

K is an empirically determined constant, for instance 40;

charge rate and transfer rate are determined from the prediction functions;

storage charge is a parameter which expresses a 'cost per byte' of disk space (default is 1);

delay tolerance expresses the relative importance to the user of rapid response to requests to access this document (default is 1);

P is normalised so that $P = 1$ for an estimated access frequency of 1/hour. The agent 62 keeps a record of the times of the last few user requests to view the document (this is what is meant by an 'access') in order to calculate this parameter. So, if $P = 1$, charge rate = 0.3, transfer rate (low) = $4293.41 * 0.0803303$, size = 10KB, storage charge = delay tolerance = 1:

$$\theta = 0.4 * 10^{-8} * 0.3 / (4293.41 * 0.0803303)^2 * 10000 = 0.36$$

$$CRP = (1 - \exp(-0.36)) / (1 + \exp(-0.36)) = 0.47$$

which is a moderate value so that there is a fair chance the document will be retained in the cache, whereas, for the peak transfer rate,

$$\theta = 0.4 * 10^{-8} * 0.3 / (4293.41)^2 * 10000 = 0.99$$

$$CRP = (1 - \exp(-0.99)) / (1 + \exp(-0.99)) = 0.03$$

which is a low value so that the document is likely to be deleted when the cache becomes full. A larger document such as 100KB, with the low transfer rate gives:

$$\theta = 0.4 * 10^{-8} * 0.3 / (4293.41 * 0.0803303)^2 * 100000 = 3.6$$

$$CRP = (1 - \exp(-3.6)) / (1 + \exp(-3.6)) = 1.0$$

and so the document is almost certain to be retained. At the higher rate, the CRP is still small, even with the larger document.

CLAIMS

1. A distributed information system comprising a network, a plurality of sites containing information and having access to the network, a first user having access to the network, and means for determining the value or change in value of the system to the first user in response to a proposed modification of the distribution of the information in the system, wherein the first user is arranged to make the proposed modification if the value or change in value to the first user exceeds a threshold.
2. A system as claimed in Claim 1, wherein the user is at least one of the plurality of sites.
3. A system as claimed in Claim 1 or Claim 2, wherein at least one of the plurality of sites includes a local data store.
4. A system as claimed in any one of the preceding claims, wherein the value of the system to the user includes the intrinsic value of the information.
5. A system as claimed in any one of the preceding claims, wherein the value of the system to the user includes an estimated cost factor to the user of the proposed modification.
6. A system as claimed in any one of the preceding claims, wherein the value of the system to the user includes an estimated time delay factor associated with the proposed modification.

7. A system as claimed in any one of the preceding claims, comprising a second user having access to the network and means for communication between the first user and the second user.
8. A system as claimed in any one of the preceding claims, in which the first user comprises means for generating a model of the distribution of information in the system.
9. A system as claimed in Claim 8, comprising means for modifying the model of the distribution.
10. A system as claimed in any one of the preceding claims, wherein the proposed modification comprises movement of the information.
11. A system as claimed in Claim 3 or in any one of Claims 4 to 9 when dependent on Claim 3, wherein the proposed modification comprises deletion of the information from the local data store.
12. A system as claimed in any one of the preceding claims, comprising means to evaluate a request from another user.
13. A system as claimed in any one of the preceding claims, comprising means to formulate a request to another user.
14. A system as claimed in any one of the preceding claims, in which the determining means comprises means for representing at least one parameter of information transfer among the sites via the network, means for requesting from the representing means a parameter value of

the at least one parameter for a specified information transfer between a first and a second of the sites via the network, and means responsive to the parameter value for forming the value or change in value of the system.

15. A system as claimed in Claim 14, in which the at least one parameter is a function of time of day.

16. A system as claimed in Claim 14 or 15, in which the at least one parameter is a function of day of the week.

17. A system as claimed in any one of Claims 14 to 16, in which the at least one parameter comprises rate of transfer.

18. A system as claimed in any one of Claims 14 to 17, in which the at least one parameter comprises cost of transfer.

19. A system as claimed in any one of Claims 14 to 18, in which the determining means further comprises means for assessing a preceding information transfer and for updating the representing means.

20. A distributed information system comprising a network, a plurality of sites containing information and having access to the network, means for representing at least one parameter of information transfer among the sites via the network, means for requesting from the representing means a value of the at least one parameter for a specified information transfer between a first and a second of the sites via the network, and means responsive to the value or change in value for taking a predetermined action relating to the specified information transfer.

21. A system as claimed in Claim 20, in which the at least one parameter is a function of time of day.
22. A system as claimed in Claim 20 or 21, in which the at least one parameter is a function of day of the week.
23. A system as claimed in any one of Claims 20 to 22, in which the at least one parameter comprises rate of transfer.
24. A system as claimed in any one of Claims 20 to 23, in which the at least one parameter comprises cost of transfer.
25. A system as claimed in any one of Claims 20 to 24, in which the action taking means is arranged to take the predetermined action if the value or change in value is within a predetermined range.
26. A system as claimed in any one of Claims 20 to 26, in which the predetermined action comprises displaying the value.
27. A system as claimed in any one of Claims 20 to 26, in which the predetermined action comprises instituting the specified information transfer.
28. A system as claimed in any one of Claims 20 to 25, in which the predetermined action comprises deleting information from a cache.
29. A system as claimed in any one of Claims 20 to 26, in which the predetermined action comprises ascertaining a preferred time for the

specified information transfer and instituting the specified information transfer at the preferred time.

30. A system as claimed in any one of Claims 20 to 29, further comprising means for assessing a preceding information transfer and for updating the representing means.

31. A distributed information system substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.



Application No: GB 9619043.4
Claims searched: 1-31

Examiner: Al Strayton
Date of search: 4 December 1996

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A: AUSB; AUXF. H4K: KF42; KOD4; KOT

Int Cl (Ed.6): G06F, H04Q

Other: ONLINE: WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	GB 2 298 299 A (MELNIKOFF) See the abstract	1,20 at least
X	GB 2 294 788 A (REUTERS) See the abstract	"
X	GB 2 294 132 A (GEC-MARCONI) See the abstract	"
A	WO 96/25012 A1 (BT)	
A	WO 96/23265 A1 (BT)	
X	WO 96/18963 A1 (FS HOLDINGS) See the abstract	1,20 at least
X	WO 96/05563 A1 (REUTERS) See the abstract	"
A	WO 95/30317 A1 (BT)	
X	US 5 305 200 (HARTHEIMER...)	1,20 at least
X	US 5 202 827 (SOBER)	"

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.

& Member of the same patent family

A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.